

REMARKS

This Amendment is submitted in response to the Office Action dated August 1, 2005, having a shortened statutory period set to expire November 1, 2005. Proposed amendments include amending Claims 1, 2, 8 and 12, cancelling Claim 9, and adding Claims 16-17. Upon entry of the proposed amendments, Claims 1-8 and 10-17 will be pending.

Applicant appreciates the time and courtesy extended by the Examiner during a February 27, 2006 telephone interview. No formal agreement, other than that described below for the 102 rejection, was reached during this teleconference. If the Examiner believes that further discussion would be useful in the prosecution of the present application, a telephone call to the undersigned representative at 512.617.5533 would be greatly appreciated.

Objections to the Drawings

In paragraph 2 of the present Office Action, the Examiner has objected to Figures 6 and 7 for failing to contain the legend "Prior Art." Attached are corrected drawings, which contain no new matter.

Rejections under 35 U.S.C. § 102

In paragraph 4 of the present Office Action, Claims 1, 8 and 10 are rejected under 35 U.S.C. § 102(e) as being anticipated by *Price et al.* (U.S. Patent Publication No. 2002/0085224 – "Price"). It is the understanding of the undersigned that the current amendments overcome this rejection, as the cited art does not teach or suggest "wherein the menu screen is displayed on a same browser screen that is displaying the PDF file, and wherein the menu screen displays the plurality of PDF file names with their respective address data." Thus, Applicants respectfully request that this rejection be withdrawn.

Rejections Under 35 U.S.C. § 103

In paragraph 6 of the present Office Action, Claims 1, 3, 5 and 7 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi et al.* (U.S. Patent Publication No. 2001/0049703 – "Miyoshi") in view of *Bell et al.* (U.S. Patent No. 5,301,321 – "Bell"). In

paragraph 7 of the present Office Action, Claim 2 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi* and *Bell* in further view of *Berstis* (U.S. Patent No. 6,018,345 – “*Berstis*”). In paragraph 4 of the present Office Action, Claim 4 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi* and *Bell* in further view of *Otsuka* (U.S. Patent No. 5,579,126 – “*Otsuka*”). In paragraph 9 of the present Office Action, Claim 6 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi* and *Bell* in further view of *Mical* (U.S. Patent No. 4,772,882 – “*Mical*”). In paragraph 10 of the present Office Action, Claims 8-10, 12, 14 and 15 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi* in view of *Mical*. In paragraph 11 of the present Office Action, Claims 11 and 13 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Miyoshi* in view of *Berstis*. Applicants respectfully traverse these rejections in light of the proposed amendments.

Miyoshi teaches a method for printing a Web page based on a “Read-Ahead” program, which sends control numbers (which are numbers that are assigned to particular Web pages to be printed – see paragraph [0039]) with Web pages to a printer (see paragraph 0043)).

Bell is cited for the known function of initializing a program with a click of a mouse button.

Berstis teaches changing a mouse pointer to a “Go” pointer when the original mouse pointer is positioned over a hot link on a Web page.

Mical is cited for teaching the use of a menu list, and for reading a menu by clicking a mouse.

With regards to exemplary **Claim 1**, the cited art does not teach or suggest “the menu screen is displayed on a same browser screen that is displaying the PDF file, and wherein the menu screen displays the plurality of PDF file names with their respective address data,” as supported in the present application at, *inter alia*, on Page 8 and Figure 4b. While *Miyoshi* teaches that a listing of files to be printed can be displayed on a monitoring screen 180a, this

screen is not the same as the screen on which the PDF file is currently displayed. (See paragraph [0043] of *Miyoshi*, which may even be broken apart and displayed in two separate screens. There is no suggestion that the menu screen is displayed on the same browser screen that is displaying the PDF file, thus avoiding toggling back and forth between the PDF screen and the print queue screen, as is required by *Miyoshi*. Similarly, the feature (which is supported in the specification at, *inter alia*, the last paragraph of page 8) found in new **Claim 16** of the menu screen being “superimposed on the browser screen, and wherein the menu screen contains multiple network addresses that are to be sent to a print queue” is not taught or suggested by the cited art.

With regards to **Claim 2**, the cited art does not teach or suggest a data control unit that “changes the mouse pointer to a PDF file specific form when it has determined that said file name data is of a PDF file.” *Miyoshi* teaches a changed pointer icon (“Go? Cursor”) that appears when the original pointer is over an active link site (*Miyoshi*, col. 6, lines 1-4). *Miyoshi* does not try to describe what type of link (i.e., to a PDF file) may be activated. That is, while *Miyoshi* provides a visual prompt to the user to click a “GO” button that is “on the remote control and possibly on the keyboard as well” (*Miyoshi*, col. 6, lines 6-7), The “GO? cursor” does not identify what type of document link is involved. As the Examiner correctly states, the link may be to an HTML link or another format link, but there is no teaching or suggestion that *Miyoshi*’s “Go? cursor” lets the user know what type (format) link is being presented. Furthermore, the cited art does not teach or suggest the feature of “an activation by the mouse pointer of the PDF file specific form causes the menu screen to replace a display of the PDF file specific form on the browser screen,” as supported in the specification on page 11, lines 4-8 of the specification. That is, the cited art does not teach or suggest that clicking the PDF file specific form (the new looking mouse pointer) causes that new mouse pointer to disappear on the screen, to be replaced with the menu screen showing the PDF file names.

Regarding amended **Claim 8** (which now incorporates the features of cancelled **Claim 9**), the cited art does not teach or suggest “and if the pull printing is not indicated, the PDF direct print menu screen is read out by clicking the mouse while the mouse pointer is kept specifying the file, the specified PDF file is downloaded and saved in the network terminal device by

selecting a menu item for saving an object in a file on the PDF direct print menu screen, and said specified PDF file is outputted to the printer with a PDF direct printing function.” *Miyoshi* is cited in the present Office Action for teaching this feature through the use of “default saving button 182.” However, default saving button 182 “is used when the print format is set or updated. When the print format is set and then the default saving button 182 is clicked, a newly set value is saved.” (*Miyoshi*, paragraph [0043].) That is, default saving button 182 is used to set printing formats to a default value, which is different from and does not teach or suggest reading and outputting a PDF file as claimed in newly amended Claim 8.

CONCLUSION

As the cited prior art does not teach or suggest all of the limitations of the pending claims, Applicants now respectfully request a Notice of Allowance for all pending claims.

No extension of time for this response is believed to be necessary. However, in the event an extension of time is required, that extension of time is hereby requested. Please charge any fee associated with an extension of time as well as any other fee necessary to further the prosecution of this application to **IBM CORPORATION DEPOSIT ACCOUNT No. 09-0449**.

Respectfully submitted,



James E. Boice
Registration No. 44,545
DILLON & YUDELL LLP
8911 North Capital of Texas Highway
Suite 2110
Austin, Texas 78759
512.343.6116

ATTORNEY FOR APPLICANT(S)